Jurnal Ilmu Komputer Agri-Informatika
Volume 7 Nomor 1 halaman 41 – 49
eISSN: 2654-9735, pISSN: 2089-6026

Tersedia secara *online* di:
http://journal.ipb.ac.id/index.php/jika

Pengelompokan Dokumen Tugas Akhir Mahasiswa S1 Ilmu Komputer IPB Berdasarkan Frequent Term Sets

Clustering Undergraduate Final Project of Computer Science's Student IPB Based on Frequent Term Sets

MIFTAH FARID[1*], IMAS SUKAESIH SITANGGANG[1]

Abstrak

Pengelompokan dokumen tugas akhir mahasiswa perlu dilakukan karena dokumen tugas akhir mahasiswa bertambah setiap tahunnya. Pengelompokan dokumen dilakukan agar dokumen yang memiliki kesamaan konteks dapat dikelompokkan ke dalam suatu kategori. Tujuan dari penelitian ini menerapkan teknik association rule mining (ARM) untuk menentukan frequent term sets dengan menggunakan algoritme ECLAT. Data yang digunakan dalam penelitian ini adalah data abstrak dokumen tugas akhir mahasiswa Ilmu Komputer IPB dalam bahasa Inggris. Penelitian ini menggunakan algoritme ECLAT dengan minimum support sebesar 0.1, 0.15, 0.20, 0.25, 0.30, dan 0.35. Penelitian ini menggunakan metode hierarchical frequent term based clustering untuk menentukan cluster. Frequent term sets hasil algoritme ECLAT masih terlalu umum untuk digunakan sebagai penciri dokumen. Pada penelitian ini hasil clustering dengan minimum support 0.35 terbentuk 3 tingkat hirarki term. Term yang sering muncul pada minimum support 0.35 adalah 'result', 'base', 'use', 'one', 'data'. Sedangkan asosiasi dua term yang sering muncul pada minimum support 0.35 adalah 'result-use', 'base-use', 'one-use', 'data-use'. Hasil clustering dapat mempermudah pencarian dokumen berdasarkan kata kunci tertentu.

Kata Kunci: association rule mining, ECLAT, frequent term sets, clustering hirarki.

Abstract

Clustering undergraduate final project document is necessary because undergraduate final project document increases every year. Document which has context similarity can be grouped into some category. The purpose of this study is to apply the method of association rule mining to determine frequent term sets using the ECLAT algorithm. The data used in this study are documents of final project abstract in English of undergraduate Computer Science student of IPB. This study applied the ECLAT algorithm with minimum support of 0.1, 0.15, 0.20, 0.25, 0.30, and 0.35. This study used hierarchical frequent term based clustering method for determining clusters. Frequent term sets from ECLAT algorithm is still too general to be used as an identifier of the document. Cluster with minimum support of 0.35 generates 3 level of hierarchy. The 1-frequent term sets with minimum support of 0.35 include 'result', 'base', 'use', 'one', and 'data'. The 2-frequent term sets with minimum support of 0.35 include 'result-use', 'base-use', 'one-use', and 'data-use'. The ECLAT algorithm with minimum support 0.1, 0.15, 0.20 generate 4 hierarchical levels with 3-frequent term sets as the lowest level. Clustering results can facilitate the search for documents based on certain keywords.

Keywords: association rule mining, ECLAT, frequent term sets, hierarchical clustering.

PENDAHULUAN

Dokumen tugas akhir mahasiswa bertambah setiap tahunnya. Oleh karena itu, pengelompokan dokumen tugas akhir mahasiswa perlu dilakukan. Pengelompokan dokumen

[1]Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor 16680.
*Penulis Korespondensi: Surel: miptah.farid@gmail.com

tugas akhir mahasiswa dilakukan agar dokumen yang memiliki kesamaan konteks dapat dikelompokan ke dalam suatu kategori, sehingga pencarian dokumen lebih mudah dilakukan. Pengelompokan dokumen dapat menggunakan teknik clustering. Clustering merupakan teknik data mining yang bertujuan untuk mengidentifikasi sekempok objek yang mempunyai kemiripan karakteristik tertentu yang dapat dipisahkan dengan objek lainnya sehingga objek yang berada dalam satu kelompok yang sama relatif lebih homogen dari pada objek yang berada pada kelompok yang berbeda. Menurut Han et al. (2012) clustering adalah proses pengelompokan objek ke dalam kelas yang objeknya mirip.

Ramdani (2011) telah melakukan penelitian tentang clustering dokumen berbahasa Indonesia menggunakan Bisecting K-means. Penelitian ini menggunakan konsep indexing dengan mengukur centroid maksimum dan centroid rata-rata Subandi (2014) juga telah melakukan penelitian tentang pengenalan dokumen berdasarkan abstrak dan mengelompokkan ke suatu kategori dengan menggunakan algoritme Bisecting K-means. Abstrak yang digunakan terdiri atas 78 dokumen abstrak berbahasa Indonesia dan 113 dokumen abstrak berbahasa Inggris. Hasil dari penelitian ini adalah mesin pencari dokumen. Mesin pencari dokumen pada penelitian ini memberikan keluaran berupa dokumen yang sudah dikelompokkan ke suatu kategori berdasarkan abstrak dokumen. Erman dan Sitanggang (2016) mengelompokkan dokumen abstrak bahasa Inggris dengan metode K-Means. Penelitian tersebut menggunakan Algoritme ECLAT untuk menentukan frequent itemset sebagai term pada association rule mining dan menggunakan evaluasi purity untuk mengukur kualitas clustering yang dihasilkan. Penelitian Erman dan Sitanggang (2016) tidak menerapkan konsep hirarki dalam kumpulan term pada saat pengelompokkan dokumen. Konsep hirarki dalam kumpulan term dapat digunakan untuk menentukan dokumen yang lebih spesifik.

Association rule mining digunakan untuk menemukan asosiasi yang menarik dari kumpulan data yang besar. Frequent pattern/ frequent itemset adalah pola yang muncul berkali-kali dalam suatu data. Menemukan frequent pattern menjadi peranan penting dalam associations mining, korelasi, dan banyak hubungan menarik lainnya dalam data. Selain itu, frequent pattern membantu dalam klasifikasi data, clustering, dan pekerjaan data mining lainnya (Han et al. 2012).

Penelitian ini bertujuan untuk 1) menerapkan metode association rule mining untuk menentukan frequent term sets dari dokumen abstrak tugas akhir mahasiswa S1 Program Studi Ilmu Komputer IPB, 2) mengelompokkan dokumen tugas akhir mahasiswa S1 Program Studi Ilmu Komputer IPB berdasarkan frequent term sets dan hirarki dari frequent term sets. Pengelompokan dokumen berdasarkan hirarki frequent term dilakukan menggunakan algoritme Hierarchical Frequent Term based Clustering (HFTC). Hirarki frequent term dapat digunakan untuk menentukan cluster dokumen yang lebih spesifik terkait beberapa kata kunci tertentu. Manfaat dari penelitian ini adalah kelompok dokumen yang dapat mempercepat proses pencarian dokumen tugas akhir mahasiswa S1 Program Studi Ilmu Komputer IPB.

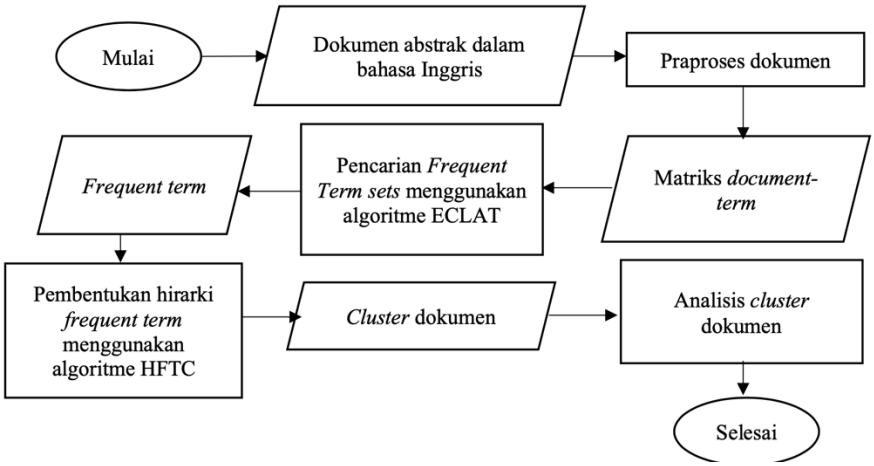
METODE

Data Penelitian

Data yang digunakan dalam penelitian ini adalah abstrak dari dokumen tugas akhir mahasiswa S1 Program Studi Ilmu Komputer IPB dalam bahasa Inggris tahun 2012 sampai 2014 sebanyak 346 dokumen yang dinyatakan dalam format txt. Data yang digunakan dalam penelitian ini diperoleh oleh peneliti sebelumnya (Erman dan Sitanggang 2016).

Tahapan Penelitian

Tahapan penelitian dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1 Tahapan penelitian

Praproses Dokumen

Pada tahapan praproses dokumen dilakukan beberapa tahapan, yaitu *lowercasing,* pembuangan tanda baca dan angka, pembuangan *stopwords*, pembuangan *whitespace, stemming* dan pembuatan matriks *document-term.*

- *Lowercasing. Lowercasing* adalah proses mengubah semua huruf menjadi huruf kecil. Proses ini dilakukan agar setiap term pada dokumen menjadi *case-sensitif* pada saat pemrosesan teks dokumen. Pada penelitian ini semua huruf dalam abstrak dokumen tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB diubah menjadi huruf kecil.
- Pembuangan tanda baca dan angka. Data abstrak yang sudah dilakukan proses *lowercasing* selanjutnya dilakukan proses pembuangan tanda baca dan angka. Proses ini dilakukan agar tanda baca dan angka yang tidak berhubungan dengan pengelompokan dokumen tidak mempengaruhi hasil.
- Pembuangan *stopwords*. *Stopwords* adalah daftar kata-kata yang tidak memiliki makna. *Stoptlist/stopwords* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* (Feldman dan Sanger 2007). Pada umumnya kata-kata *stopwords* memiliki frekuensi kemunculan yang tinggi di setiap dokumen. Pada penelitian ini *stopwords* yang digunakan tersedia pada *package* tm di pemrograman R. Pada penelitian ini juga menggunakan *stopwords* tambahan yang sering muncul pada dokumen tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB.
- Pembuangan *whitespace*. Tahapan pembuangan *whitespace* dilakukan untuk menghilangkan karakter tabulasi(*tab*), spasi (*space*), *new line (enter)* pada data abstrak dokumen tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB.
- *Stemming*. *Stemming* adalah menentukan kata dasar dari setiap *term* dengan cara menghilangkan imbuhan.
- Pembuatan matriks *document-term*. Matrixs *document-term* adalah matriks dengan *term* sebagai baris dan dokumen sebagai kolom yang menggambarkan frekuensi kemunculan *term* pada dokumen. Gambaran matriks *document-term* ditunjukkan pada Gambar 2.

	$term_1$	$term_2$	...	$term_n$
doc_1	$freq_{11}$	$freq_{12}$		$freq_{1n}$
doc_2	$freq_{21}$	$freq_{22}$		$freq_{2n}$
...				...
doc_m	$freq_{m1}$	$freq_{m2}$	...	$freq_{mn}$

Gambar 2 Matriks *document-term*

Pencarian *Frequent Term Sets* Menggunakan Algoritme ECLAT

Algoritme ECLAT merupakan yang pertama mengusulkan untuk menghasilkan semua *frequent term sets* dengan cara *depth-first*, tetapi menggunakan *layout database* dan pendekatan titik potong untuk menentukan *minimum support dari* sebuah *itemset*. *Pseudocode* algoritme ECLAT ditunjukkan pada Gambar 3. Pada penelitian ini algoritme ECLAT digunakan untuk menentukan *frequent term sets.*

Algoritme 1: ECLAT – Frequent Itemset Mining	
Input:	Database transaksi D, Threshold min_{sup} yang ditentukan pengguna Kumpulan term S
Output:	Frequent itemsets F
Prosedur: Eclat(S)	1. Untuk semua term $A_i \in S$ 2. $T_i=\emptyset$ # T_i: kumpulan frequent itemset 3. Untuk semua term $A_j \in S$, dengan $j > i$ lakukan 4. $R= A_i \cup A_j$; 5. $L(R)=L(A_i) \cap L(A_j)$; 6. *Jika* $support(R) \geq min_{sup}$ *maka* 7. $T_i=T_i \cup \{R\}$; 8. $F_{\|R\|}=F_{\|R\|} \cup \{R\}$; 9. selesai 10. selesai 11. selesai 12. untuk semua $T_i \neq \emptyset$ lakukan Eclat(T_i);

Gambar 3 *Pseudocode* algoritme ECLAT (Guandong *et al.* 2010)

Pembentukan Hirarki *Frequent Term Sets* Menggunakan Algoritme HFTC

Algoritme *Hierarchical Frequent Term based Clustering* (HFTC) menentukan sebuah hirarkikal *clustering* dengan himpunan kosong *frequent term sets* pada *root* hirarki. Algoritme HFTC menggunakan *1-frequent term sets* di tingkat pertama hirarki, dan *2-frequent terms sets* di tingkat berikutnya. Algoritme HFTC berhenti menambah tingkat pada hirarki ketika tidak ada *n-frequent term sets* untuk tingkat selanjutnya (Beil *et al.* 2002). *Pseudocode* algoritme HFTC ditunjukkan pada Gambar 4. Pada penelitian ini algoritme HFTC digunakan untuk menentukan *cluster* dan pembuatan hirarki *cluster. Frequent term sets* yang didapat dari algoritme ECLAT selanjutnya dihitung *minimum overlap* untuk setiap set. *Minimum overlap* digunakan untuk menentukan *frequent term sets* sebagai kandidat *cluster.*

```
HFTC (basis data D, float minsup)

1)  TermSetsTerpilih := {} ;
2)  N:= |D|;
3)  TermSetsSisa := FreqTermsets (D, minsup);
4)  S:= TermSetsSisa
5)   selama | cov(TermSetsTerpilih) ≠ n, lakukan
6)            Untuk setiap set dalam S, lakukan
                hitung himpunan overlap;          selesai

      buang semua dokumen dalam cov(CandidateTerbaik) dari D dan dari TermSetsSisa;

7)          CandidateTerbaik := elemen S dengan minOVerlap;
    TermSetsTerpilih := TermSetsTerpilih ∪  {CandidateTerbaik};

8)          S := S – {CandidateTerbaik};
9)     buang semua dokumen dalam cov(CandidateTerbaik) dari D dan dari TermSetsSisa;
10) kembalikan TermSetsTerpilih dan cov(TermSetsTerpilih);
```

Gambar 4 *Pseudocode* algoritme HFTC (Beil *et al.* 2002)

Analisis *Cluster* Dokumen

Tahapan ini menganalisis *cluster* dokumen untuk mengukur kualitas *cluster* yang dibentuk dari hirarki *frequent term sets*. Pada tahap ini membahasa tentang *cluster* yang dihasilkan berdasarkan *frequent term sets* yang membentuk *n-frequent term sets* yang berada di tingkat yang lebih tinggi pada hirarki *cluster.*

Lingkungan Pengembangan

Penelitian ini dilakukan menggunakan perangkat keras berupa komputer dengan spesifikasi sebagai berikut prosesor Intel® Core™, *memory* 4 GB, dan 750 GB HDD. Perangkat lunak yang digunakan adalah sistem operasi Windows 10, R sebagai bahasa pemrograman yang digunakan untuk mengolah data teks, Microsoft Visio, dan Notepad++ sebagai *text editor.*

HASIL DAN PEMBAHASAN

Praproses Dokumen

Pada penelitian ini dokumen abstrak yang digunakan untuk data berjumlah 346 dokumen dalam format fail txt. Tahap praproses data dimulai dengan tahapan *lowercasing*. Setelah dilakukan tahapan *lowercasing*, dilakukan praproses selanjutnya yaitu pembuangan tanda baca dan angka. Tahap ketiga adalah pembuangan *stopwords*. Pada tahap ini dilakukan penambahan *stopwords* yang diperoleh dari penelitian Erman dan Sitanggang (2016). Beberapa contoh *stopwords* tambahan adalah ‘algorithm’, ‘browser’, ‘computer’, ‘input’.

Setelah pembuangan *stopwords* dilakukan pembuangan *whitespace* dan *stemming*. Pada hasil tahap *stemming* ada beberapa *term* yang berubah menjadi kata yang tidak terdapat di dalam kamus bahasa Inggris. Beberapa hasil dari tahapan *stemming* ditunjukkan pada Tabel 1.

Tabel 1 Contoh hasil tahapan *stemming*

Sebelum tahap *stemming*	Setelah tahap *stemming*	Kata seharusnya
classification	classifi	Classify
identification	identifi	Identify
queries	queri	Query

Setelah *stemming* tahapan praproses data selanjutnya adalah pembuatan matriks *document-term*. Matriks *document-term* adalah matriks dengan *term* sebagai baris dan dokumen sebagai kolom yang menggambarkan frekuensi kemunculan *term* pada dokumen. Matriks yang terbentuk dari proses ini berukuran 346 × 3493. Untuk mengurangi dimensi matriks, dilakukan pembuangan *term* yang memiliki frekuensi rendah. Proses ini dilakukan dengan menggunakan fungsi *removeSparseTerm()* pada bahasa pemrograman R. Nilai *sparse* yang digunakan adalah sebesar 0.95. Nilai *sparse* yang digunakan sesuai dengan nilai *sparse* yang digunakan pada penelitian Erman dan Sitanggang (2016). *Term* dengan frekuensi rendah dibuang untuk mereduksi dimensi matriks *document-term*.

Pencarian *Frequent Term Sets* Menggunakan Algoritme ECLAT

Proses ini mencari *frequent term sets* dari *term* pada dokumen abstrak. Contoh *frequent term sets* dapat dilihat pada Tabel 2. Tahap ini dilakukan menggunakan algoritme ECLAT yang tersedia pada *package arules* dalam Bahasa pemrograman R. Algoritme ECLAT yang dijalankan menggunakan parameter *minimum support* sebesar 0.1, 0.15, 0.2, 0.25, 0.3, 0.35.

Tabel 2 Perbandingan jumlah *term* dan *frequent term sets* berdasarkan nilai *minimum support.*

Minimum support	Jumlah *term*	Jumlah *frequent term sets*
0.10	85	271
0.15	37	90
0.20	18	36
0.25	14	24
0.30	8	15
0.35	8	12

Tabel 2 menunjukkan bahwa semakin tinggi nilai *minimum support* semakin sedikit jumlah *term* dan *frequent term sets* yang dihasilkan. Hal ini dikarenakan nilai *minimum support* mencari kemunculan *frequent term sets* di dalam keseluruhan matriks *document term.* Tabel 3 menunjukkan bahwa 2-*frequent term sets* merupakan kombinasi dari 1-*frequent term sets,* dan 3-*frequent term sets* merupakan kombinasi dari 2-*frequent term sets*. Sebagai contoh *frequent term sets* 'result-show-use' dihasilkan dari *frequent term sets* 'result-use' dan 'result-show'.

Tabel 3 Contoh *frequent term sets* hasil algoritme ECLAT dengan *minimum support* 0.15

1-*frequent term sets*	2-*frequent term sets*	3-*frequent term sets*
Show	Base-data	Data-base-use
Use	Resutl-use	Data-develop-use
Data	Result-show	Result-show-use
result	Data-develop	Data-show-use

Tabel 4 menunjukkan bahwa semakin tinggi nilai minimum support maka semakin sedikit *frequent term sets* yang diperoleh. Hal ini terjadi karena nilai *minimum support* mencari kemunculan *frequent term sets* dalam keseluruhan matriks *document-term. Minimum support* 0.35 mempunyai arti *frequent term sets* yang muncul sebanyak 35% dalam matriks *document-term.*

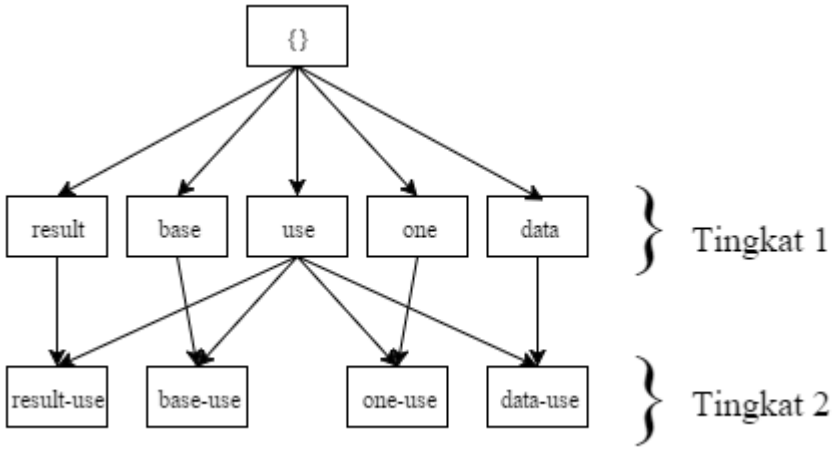
Table 4 *Term* dan *frequent term sets* yang dihasilkan algoritme ECLAT dengan nilai *minimum support* 0.20, 0.25, 0.30, 0.35.

Minimum support	*Term*	*Frequent term sets*
0.20	Use, data, one, base, result, show,develop, inform, studi, perform, test, time, obtain, valu, implement, feature, aim, user	Feature-use, implement-use, use-valu, obtain-use, time-use, test-use, perform-use, studi-use, inform-use, develop-use, show-use, result-show, result-use, base-use, one-use, data-one, data-use, result-show-use
0.25	Use, data, one, base, result, show, develop, inform, studi, perform, test, time, obtain, valu	Test-use, perform-use, studi-use, inform-use, develop-use, show-use, result-use, base-use, one-use, data-use
0.30	Use, data, one, base, result, show, develop, inform	inform-use, develop-use, show-use, result-use, base-use, one-use, data-use
0.35	Use, data, one, base, result, show, develop, inform	result-use, base-use, one-use, data-use

Pembentukan Hirarki *Frequent Term Sets* Menggunakan Algoritme HFTC

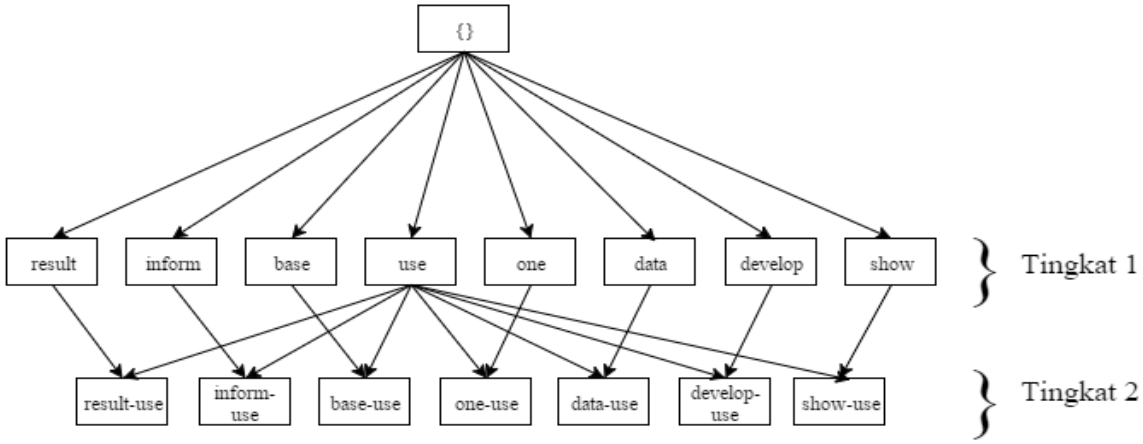
Pada pembentukan hirarki *frequent term sets* dimulai dengan himpunan kosong sebagai *root.* Pada tingkat 1 dibentuk dari kumpulan *1-frequent term sets* dari hasil algoritme ECLAT. Pada tingkat 2 dari hirarki tidak semua kombinasi *term* menjadi *2-frequent term sets*. *Frequent term sets* pada tingkat 2 hirarki diperoleh dari hasil algoritme ECLAT. Contoh hirarki dari *frequent term sets* untuk *minimum support* 0.35 dan 0.30 dapat dilihat pada Gambar 5 dan Gambar 6.

Gambar 5 menunjukkan *cluster* yang terbentuk dari *frequent term sets* 'result', 'base', 'use', 'one', 'data' dengan *minimum support* 0.35. Pada tingkat 1 hirarki terdapat 5 *cluster,* sedangkat pada tingkat 2 hirarki terdapat 4 *cluster* dengan kombinasi *term-term* pada tingkat 1 hirarki.



Gambar 5 Hirarki dari *frequent term sets* untuk hasil algoritme ECLAT dengan *minimum support* 0.35

Gambar 6 menunjukkan pada tingkat 1 hirarki terbentuk 8 *cluster* dengan *minimum support 0.30.* Tingkat 2 *cluster* terbentuk 7 *cluster* dari kombinasi *term-term* pada tingkat 1 hirarki.



Gambar 6 Contoh hirarki dari *frequent term sets* untuk hasil algoritme ECLAT dengan *minimum support* 0.30.

Analisis *Cluster* Dokumen

Pada 2-*frequent term sets* dokumen yang mengandung kedua kombinasi *term* yang membentuk 2-*frequent term sets*. Pada *term* ‘result-show-use’ diperoleh dari kombinasi *term* ‘result-show’, ‘show-use’, ‘result-use’, sehingga dokumen-dukumen yang mengandung *term* ‘resull-show-use’ diperoleh dari seleksi dokumen-dokumen yang mengandung *term* ‘result-show’, ‘show-use’, ‘result-use’. Dokumen-dokumen yang dipilih adalah dokumen yang *overlap* antara *term* ‘result-show’, ‘show-use’, ‘result-use’.

Hasil algoritme ECLAT dengan *minimum support* 0.20 terdapat 1 *cluster* pada tingkat 3 hirarki dengan *term* ‘result-show-use’. Pada 3-*frequent term sets* ‘result-show-use’ diperoleh dari kombinasi *frequent term sets* ‘result-show’, ‘result-use’, ‘show-use’. Pada *term* ‘result-show-use’ diperoleh *cover* dokumen yang mengandung *term* tersebut adalah {6,8,13,21,23,29,30,...,344}. *Cover* suatu *frequent term sets*(S) adalah kumpulan dokumen yang mengandung *term* pada *frequent term sets*(S) (Beil *et al.* 2002). Cover dokumen *term* ‘result-show-use’ dapat dikelompokkan menjadi 1 *cluster.*

Tabel 6 Dokumen yang *overlap* dari kumpulan *term* hasil algoritme ECLAT dengan *minimum support* 0.20.

Term	Dokumen-ID	2-*frequent term sets*	Dokumen-ID
Result	4, 6, 7, 8, 9, 10, 13, 14, 21, 23, 27, 28, 29, 30, 32, 33, 35, …, 344	*Result-show*	6, 8, 13, 21, 23, 29, 30, 32, 37, 42, 48, 49, 51, 54, 59, 61, 65, …, 344
Show	6, 8, 13, 21, 23, 24, 26, 29, 30, 32, 37, 38, 41, 42, 43, 45, 48, …, 344	*Show-use*	6, 8, 13, 21, 23, 24, 26, 29, 30, 32, 37, 38, 41, 42, 43, 45, 48, …, 344
Use	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, …, 346.	*Result-use*	4, 6, 7, 8, 10, 13, 14, 21, 23, 27, 28, 29, 30, 32, 33, 35, 37, …, 344
Data	1, 2, 3, 4, 8, 11, 12, 14, 15, 16, 18, 19, 21, 27, 30, 31, 33, …, 345	*Base-data*	11, 12, 15, 19, 21, 27, 30, 33, 36, 42, 44, 45, 46, 49, 56, 60, 63, …, 345
Base	6, 7, 9, 11, 12, 13, 15, 17, 19, 20, 21, 22, 25, 26, 27, 29, 30, …, 346		

SIMPULAN

Dari penelitian ini dapat diambil kesimpulan bahwa pada algoritme ECLAT semakin besar nilai *minimum support* yang digunakan maka semakin sedikit *frequent term sets* yang dihasilkan. Pada penelitian ini hasil *frequent term sets* dari algoritme ECLAT masih terlalu umum untuk dijadikan kata penciri untuk pengelompokan dokumen tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB. Pada penelitian ini dengan *minimum support* 0.35 tidak terdapat *cluster* pada tingkat 3 dalam hirarki *frequent term*. *Cluster* pada tingkat 3 dalam hirarki yaitu 3-*frequent term sets* diperoleh pada *minimum support* 0.10, 0.15, 0.20. *Term-term* yang mewakili dokumen abstrak tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB diantaranya 'result-show-use' dengan *minimum support* 0.20.

Penelitian ini masih memiliki kekurangan yaitu kurang optimalnya proses penentuan *frequent term sets* dilihat dari kumpulan *frequent term sets* yang masih terlalu umum. Saran untuk penelitian selanjutnya adalah menentukan nilai *sparse* yang sesuai untuk mendapatkan *term-term* yang mewakili dokumen abstrak penelitian tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB. Penelitian selanjutnya dapat menggunakan data yang lebih lengkap dari data abstrak dokumen. Pada tahapan pembuangan *stopwords* terdapat kekurangan yaitu kurang lengkapnya *stopwords* yang digunakan. Saran untuk penelitian selanjutnya adalah menambah *stopwords* terkait dokumen tugas akhir mahasiswa S1 program studi Ilmu Komputer IPB.

DAFTAR PUSTAKA

Beil F, Ester M, Xu X. 2002. Frequent term-based text clustering. Di dalam: *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002 Jul. hlm 436-442.

Erman L M, Sitanggang I S. 2016. Clustering undergraduate computer science student final project based on frequent itemset. *International Journal of Information Technology and Computer Science* (IJITCS). 8(11): 1-7.

Feldman R, Sanger J. 2007. *The Text Mining Handbook.* Cambridge (UK): Cambridge University Press.

Han J, Kember M, Pei J. 2012. *Data Mining Concepts and Techniques Ed ke-3*. Waltham (US): Morgan Kaufmann Publisher.

Guandong X, Yanchun Z, Lin L. 2010. *Web Mining and Social Networking: Techniques and Applications.* New York (US): Spring Science & Business Media.

Ramdani H. 2011. *Clustering* konsep dokumen berbahasa indonesia menggunakan *bisecting k-means* [skripsi]. Bogor (ID): Institut Pertanian Bogor.

Subandi N.A. 2014. *Clustering* dokumen skripsi berdasarkan abstrak dengan menggunakan *bisecting k-means* [skripsi]. Bogor (ID): Institut Pertanian Bogor.